Notice of Allowability	Application No.	Applicant(s)	
	09/982,395	SKJOLSVOLD, ARILD E.	
	Examiner	Art Unit	
	Dennis G. Bonshock	2173	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address--

All claims being allowable, PROSECUTION ON THE MERITS IS (OR REMAINS) CLOSED in this application. If not included herewith (or previously mailed), a Notice of Allowance (PTOL-85) or other appropriate communication will be mailed in due course. **THIS NOTICE OF ALLOWABILITY IS NOT A GRANT OF PATENT RIGHTS.** This application is subject to withdrawal from issue at the initiative of the Office or upon petition by the applicant. See 37 CFR 1.313 and MPEP 1308.

1. ☒ This communication is responsive to *the Applicant's Amendment filed on 3-16-07 and the Examiner Amendment of 5-9-07*.

2. ☒ The allowed claim(s) is/are *1-6, 8-11, 13-15, 17-20, 22-28, and 40*.

3. ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
   a) ☐ All b) ☐ Some* c) ☐ None of the:
      1. ☐ Certified copies of the priority documents have been received.
      2. ☐ Certified copies of the priority documents have been received in Application No. _____ .
      3. ☐ Copies of the certified copies of the priority documents have been received in this national stage application from the International Bureau (PCT Rule 17.2(a)).

   * Certified copies not received: _____.

Applicant has THREE MONTHS FROM THE "MAILING DATE" of this communication to file a reply complying with the requirements noted below. Failure to timely comply will result in ABANDONMENT of this application.
THIS THREE-MONTH PERIOD IS NOT EXTENDABLE.

4. ☐ A SUBSTITUTE OATH OR DECLARATION must be submitted. Note the attached EXAMINER'S AMENDMENT or NOTICE OF INFORMAL PATENT APPLICATION (PTO-152) which gives reason(s) why the oath or declaration is deficient.

5. ☐ CORRECTED DRAWINGS (as "replacement sheets") must be submitted.
   (a) ☐ including changes required by the Notice of Draftsperson's Patent Drawing Review (PTO-948) attached
      1) ☐ hereto or 2) ☐ to Paper No./Mail Date _____.
   (b) ☐ including changes required by the attached Examiner's Amendment / Comment or in the Office action of Paper No./Mail Date _____.

   Identifying indicia such as the application number (see 37 CFR 1.84(c)) should be written on the drawings in the front (not the back) of each sheet. Replacement sheet(s) should be labeled as such in the header according to 37 CFR 1.121(d).

6. ☐ DEPOSIT OF and/or INFORMATION about the deposit of BIOLOGICAL MATERIAL must be submitted. Note the attached Examiner's comment regarding REQUIREMENT FOR THE DEPOSIT OF BIOLOGICAL MATERIAL.

Attachment(s)

1. ☐ Notice of References Cited (PTO-892)
2. ☐ Notice of Draftperson's Patent Drawing Review (PTO-948)
3. ☐ Information Disclosure Statements (PTO/SB/08), Paper No./Mail Date _____
4. ☐ Examiner's Comment Regarding Requirement for Deposit of Biological Material
5. ☐ Notice of Informal Patent Application
6. ☐ Interview Summary (PTO-413), Paper No./Mail Date _____ .
7. ☒ Examiner's Amendment/Comment
8. ☒ Examiner's Statement of Reasons for Allowance
9. ☐ Other _____.

EXAMINER'S AMENDMENT

1. An examiner's amendment to the record appears below. Should the changes and/or additions be unacceptable to applicant, an amendment may be filed as provided by 37 CFR 1.312. To ensure consideration of such an amendment, it MUST be submitted no later than the payment of the issue fee.

2. Authorization for this examiner's amendment was given in a telephone interview with Scott Gallert on 5-9-2007.

3. The application has been amended as follows:

Please replace claim 1 with:

1. (Currently Amended) A method software application, the method comprising:

injecting a dynamic link library (DLL) into an executable code of the software application being tested;

installing at least one at least one hook function into an application programming interface (API) of an operating system, the installing performed by the DLL, the at least one hook function configured to monitor for testing operating system messages communicated with the software application during execution of the software application being tested;

retrieving information descriptive of a state of operation of the software application being tested and at least one graphics element rendered during execution of the software application being tested, wherein the information identifies an executable feature associated with the at least one graphics element, and wherein at least some of the

retrieved information descriptive of the state of operation is based on messages monitored by way of the at least one hook function;

storing information related to an association between the executable feature and the at least one graphics element and the state of operation of the software application in a map data structure containing information related to at least one graphics element for testing, the association and information being stored in the map data structure during execution of the software application being tested;

automatically selecting an executable feature from the map data structure based on the association stored in the map data structure;

automatically, executing the selected executable feature associated with the graphics element; and

dynamically updating the information related to the state of operation of the software application and the association in the map data structure upon execution of the executable feature.

Please replace claim 13 with :

13. (Currently Amended) A system for generating a map, comprising:

a capture agent for retrieving information descriptive of a state of operation of a software application being tested and a plurality of graphics elements rendered during execution of the software application, the information including an executable feature associated with each graphics element, the capture agent including a dynamic link library

(DLL) configured to be injected into an executable code of the software application being tested, the DLL further configured to install at least one hook function into an application programming interface (API) of an operating system, wherein the at least one hook function is configured to monitor messages communicated between the operating system and the software application during execution of the software application being tested;

an application driver for storing information in a map data structure related to an association between each executable feature and corresponding graphics element and a state of operation of the software application during execution of the software application being tested, wherein the map data structure contains information related to at least one graphics element for testing;

an indicator for tracking a dynamic updating of the information an application driver for automatically selecting one of the executable features stored in the map data structure based on the information stored in the map data structure;

a command agent for automatically executing the selected executable feature; and

an indicator for tracking a dynamic updating of the information related to the association and the state of operation of the software application in the map data structure upon the automatic execution of the selected executable feature.

Please replace claim 22 with:

22. (Currently Amended) A method for systematically invoking an executable feature of a software application having a graphical user interface, the method comprising:

injecting a dynamic link library (DLL) into an executable code of the software application;

installing at least one hook function into an application programming interface (API) of an operating system, the installing performed by the DLL, the at least one hook function configured to monitor operating system messages communicated with the software application during execution of the software application being tested;

retrieving information descriptive of a state of operation of a software application being tested and at least one graphics element rendered during execution of the software application, the information including an executable feature associated with the at least one graphics element, at least some of the retrieved information descriptive of a state of operation of the software application retrieved by way of messages monitored by the at least one hook function;

storing information related to an association between the executable feature and corresponding graphics element and the state of operation of the software application in a map data structure to contain information related to at least one graphics element for testing, the association and information being stored in the map data structure during execution of the software application;

automatically selecting from the map data structure at least one executable feature associated with a graphics element that has not been previously executed; and

automatically executing the selected at least one executable feature

REASONS FOR ALLOWANCE

1. The following is an examiner's statement of reasons for allowance:

2. The examiner considered the Applicant's Amendment filed on 3-16-07 and the Examiner Amendment of 5-9-07 and after updated search, no other prior art of record has taught that which was presented in the amended claims

3. Therefore, claims 1-6, 8-11, 13-15, 17-20, 22-28, and 40 are allowable.

4. Independent claims 1, 13, and 22, when considered as a whole, are allowable over the prior art of record (Parker et al. and Santee et al.). Parker teaches a system that does automated testing of a GUI environment, through the generation of a mapping between GUI objects and their functions (see column 4, lines 1-26, column 16, line 53 through column 17, line 12, column 25, lines 4-8 and column 9, lines 50-67), the executing of an executable feature of the Logical Screen Element (LSE) (see column 4, lines 39-45), a LSE Manager that identifies locations of the LSEs (see column 10, line 1-9), and storing the information for GUI objects in tables in the GUI and in the memory (see column 12, lines 50-65, column 4, lines 39-45, and column 9, lines 11-21). Parker further teaches, in column 9, lines 50-67, the LSEM storing functions that correspond to (are mapped to) objects on the screen, and in column 12, lines 50-56, the test driver having access to the LSEM for driving the application. Parker further teaches the use of a hook to for monitoring an application (see column 12, lines 35-49).

5. Parker's system teaches proceeding through an order set of steps in a test script during execution, with the application progressing through different states (association mapped between script element and actual screen element, test step executed, look for change in GUI state, test next element) (see column 4, lines 38-58 and figure 5). The system receiving a function [executable feature] of a Physical Screen Element (PSE); there is then an association [mapping] (see column 17, lines 2-7) made with a Logical Screen Element (LSE) of the generic script, at runtime (see column 4, lines 21-26 and column 9, lines 59-64) for the particular step in the execution. After this element is tested (executed), the system changes which Physical Screen Element (PSE) the Logical Screen Element (LSE) is referencing [dynamically updating the association] (see column 4, lines 51-58, column 9, lines 50-67, and figure 5). Parker further teaches a test script that guides execution in a deterministic (directed manner) (see column 4, lines 38-58 and figure 5). Parker is supplemented by Santee who teaches a system for testing a user interface, via a test script, by recursively selecting screen elements traversing the UI (see column 2, lines 43-54, column 5, line 44 through column 6, line 64, and figures 6 and 8), and further teaches implementing the processing in a directed manner, either traversing the elements in a depth first or a breadth first manner (see column 6, lines 9-17).

6. However, specifically the prior art of record fails to clearly teach or support the limitations of "injecting a dynamic link library (DLL) into an executable code of the software application". And the installing a "hook function" being "performed by the DLL".

7. Any comments considered necessary by applicant must be submitted no later than the payment of the issue fee and, to avoid processing delays, should preferably accompany the issue

fee. Such submissions should be clearly labeled "Comments on Statement of Reasons for Allowance."

Conclusion

8. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Dennis G. Bonshock whose telephone number is (571) 272-4047. The examiner can normally be reached on Monday - Friday, 6:30 a.m. - 4:00 p.m.

9. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John Cabeca can be reached on (571) 272-4048. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

10. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

5-10-07
dgb

JOHN CABECA
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100